

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	21	(US-20030212712-\$ or US-20040031027-\$ or US-20040062130-\$ or US-20040092255-\$ or US-20040098361-\$ or US-20040098413-\$ or US-20040098420-\$ or US-20040098421-\$ or US-20040098427-\$ or US-20050010576-\$ or US-20050010870-\$ or US-20050021572-\$ or US-20050091288-\$).did. or (US-5832520-\$ or US-6115710-\$ or US-6374250-\$ or US-6466999-\$ or US-6665787-\$ or US-6718317-\$ or US-6826626-\$ or US-6836657-\$).did.	US-PGPUB; USPAT	OR	ON	2005/09/16 18:48
L2	8	1 and index	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 18:48
L3	3489	"alignment algorithm"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:52
L5	63	"block swap"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:53
L6	1	3 and 5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:54
L7	15483	"index value"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:54

L8	15	3 and 7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:56
L9	361	order with "code block"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:57
L10	8	9 with sort\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 18:59
L11	6706	"code block"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:05
L12	9	11 near3 movement	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:00
L13	728	"code block".clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:05
L14	56	movement with sort.clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:06
L15	142	"original file".clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:06

L16	367	"new file".clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:06
L17	0	13 and 14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:06
L18	2	13 and 15	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:07
L19	2	13 and 16	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:07
L20	0	14 and 15	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:07
L21	0	14 and 16	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:08
L22	23	15 and 16	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:08
L23	1	22 and LIS	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:10

L24	12634	(707/1,2,7,100,101,202,203).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/09/16 19:11
L25	11	24 and 3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:12
L26	340	24 and 7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:12
L27	1	26 and 9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:13
L28	2	24 and 9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:13
L29	80	24 and 11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:14
L30	23	29 and movement	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:14
L31	18	30 and index	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:19

L32	1	24 and 14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/16 19:19
-----	---	-----------	---	----	----	------------------



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)Search: ☒ The ACM Digital Library ☐ The Guide**SEARCH**

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)Terms used **largest increasing subsequence**

Found 1 of 161,645

Sort results  
byDisplay  
results[Save results to a Binder](#)[Search Tips](#)☐ Open results in a new  
windowTry an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 1 of 1

Relevance scale ☐ ☐ ☐ ☐ ☐**1 Computer science in high school mathematics**Elliott I. Organick, Thomas A. Keenan, Sylviarp Charp, Alexandra Forsythe  
January 1966 **Proceedings of the 1966 21st national conference**Full text available: [pdf\(539.77 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

In the spring of 1963 an ad hoc Committee on computing chaired by Wallace Givens, and reporting to the Advisory Board of SMSG, presented a number of timely recommendations. The intent of the recommendations was to extend the mathematics program to include topics which can help explain computer organization, operation and use so that students can better relate the mathematics they have learned to the computer-influenced world around them. Many of the recommendations were rec ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **longest increasing subsequence**

Found 20 of 161,645

Sort results by

Display results


[Save results to a Binder](#)

[Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 20

Relevance scale ☐ ☐ ☐ ☐ ☐

# 1 [Session 5B: Average case analysis for batched disk scheduling and increasing subsequences](#)

E. Bachmat

 May 2002 **Proceedings of the thirty-fourth annual ACM symposium on Theory of computing**

 Full text available: [pdf\(269.32 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

(MATH) We consider the problem of estimating the tour length and finding approximation algorithms for the asymmetric traveling salesman problem arising from the disk scheduling problem. Given  $N$  requests, we show that if the seek function has positive derivative at 0 the tour length is concentrated in probability around the value  $C_{f,p} N^{1/2}$  for an explicit constant  $C_{f,p}$  dependent on the seek function and the distribution of requests. For linear ...

# 2 [How to design dynamic programming algorithms sans recursion](#)

Kirk Pruhs

 March 1998 **ACM SIGACT News**, Volume 29 Issue 1

 Full text available: [pdf\(295.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

We describe a method, which we call the Pruning Method, for designing dynamic programming algorithms that does not require the algorithm designer to be comfortable with recursion.

# 3 [Bioinformatics \(BIO\): A seriate coverage filtration approach for homology search](#)

Hsiao Ping Lee, Yin Te Tsai, Chuan Yi Tang

 March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

 Full text available: [pdf\(162.33 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The homology search within genomic databases is a fundamental and crucial work in biological knowledge discovery. With exponentially increasing size and access of databases, the issues of efficient retrieval become more essential in bioinformatics. Due to the varieties of biological data, similar sequences are not only under some error tolerance, but are also above some seriate coverage level. In this paper, we propose a seriate coverage filtration approach to extract the homologies from the dat ...

**Keywords:** homology search, incrementally decreasing covers, longest increasing subsequence problems, lossless filtration, seriate coverage

# 4 [Special section on the MAMA 2001 workshop: Average case analysis for batched disk scheduling and increasing subsequences](#)

E. Bachmat

December 2001 **ACM SIGMETRICS Performance Evaluation Review**, Volume 29 Issue 3

Full text available:  [pdf\(234.98 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

5 **EMAGEN: an efficient approach to multiple whole genome alignment**

Jitender S. Deogun, Jingyi Yang, Fangrui Ma

January 2004 **Proceedings of the second conference on Asia-Pacific bioinformatics - Volume 29 CRPIT '04**

Full text available:  [pdf\(309.96 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Following advances in biotechnology, many new whole genome sequences are becoming available every year. A lot of useful information can be derived from the alignment and comparison of different genomes. However, most of the current research focuses on pairwise genome alignment, and only a few available applications can efficiently align multiple genomes. In this paper, we present an efficient approach to align closely related multiple whole genomes, combining suffix arrays, graph theoretic formu ...

**Keywords:** cocomparability graphs, conserved regions, multiple whole genome alignment, prokaryotic genomes, suffix arrays

6 **Best sorting algorithm for nearly sorted lists**

Curtis R. Cook, Do Jin Kim

November 1980 **Communications of the ACM**, Volume 23 Issue 11

Full text available:  [pdf\(500.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Consistency control has to be enforced in database management systems (DBMS) where several transactions may concurrently access the database. This control is usually achieved by dividing the database into locking units or granules, and by specifying a locking policy which ensures integrity of the information. However, a drawback of integrity enforcement through locking policies is the degradation of the global system performance. This is mainly due to the restriction imposed by the locking ...

**Keywords:** Quicksort, Shellsort, heap-sort, insertion sort, merge sort, sorting, sorting effort measures

7 **Subset barrier synchronization on a private-memory parallel system**

Anja Feldmann, Thomas Gross, David O'Hallaron, Thomas M. Stricker

June 1992 **Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures**

Full text available:  [pdf\(1.18 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

8 **On the performance of D-redundant storage systems**

Eitan Bachmat

September 2003 **ACM SIGMETRICS Performance Evaluation Review**, Volume 31 Issue 2

Full text available:  [pdf\(234.80 KB\)](#) Additional Information: [full citation](#), [references](#)

9 **A principle of algorithm design on limited problem domain**

Jayadev Misra

June 1976 **Proceedings of the 13th conference on Design automation**

Full text available:  [pdf\(384.74 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper studies the problem of algorithm design on well defined data structures. A general principle is presented which is shown to be useful in designing algorithms which



operate on sequences (strings). A generalization of the principle is presented for more general data structures. Implications of these results are discussed.

10 A fast algorithm for computing longest common subsequences

James W. Hunt, Thomas G. Szymanski

May 1977 **Communications of the ACM**, Volume 20 Issue 5

Full text available:  [pdf\(391.38 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Previously published algorithms for finding the longest common subsequence of two sequences of length  $n$  have had a best-case running time of  $O(n^2)$ . An algorithm for this problem is presented which has a running time of  $O((r + n) \log n)$ , where  $r$  is the total number of ordered pairs of positions at which the two sequences match. Thus in the worst case the algorithm has a running time of  $O(n^2 \log n)$ . However, for those applications where most positions ...

**Keywords:** efficient algorithms, longest common subsequence

11 On the complexity of computing the measure of  $\cup[a_i, b_i]$

Michael L. Fredman, Bruce Weide

July 1978 **Communications of the ACM**, Volume 21 Issue 7

Full text available:  [pdf\(499.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The decision tree complexity of computing the measure of the union of  $n$  (possibly overlapping) intervals is shown to be  $\Omega(n \log n)$ , even if comparisons between linear functions of the interval endpoints are allowed. The existence of an  $\Omega(n \log n)$  lower bound to determine whether any two of  $n$  real numbers are within  $\epsilon$  of each other is also demonstrated. These problems provide an excellent opportunity for discussing the effects of the computational model on the ease of analysis ...

**Keywords:** analysis of algorithms, combinatorial problems, computational complexity, computational models, computational problems, decision tree programs, lower bounds

12 The Complexity of Some Problems on Subsequences and Supersequences

David Maier

April 1978 **Journal of the ACM (JACM)**, Volume 25 Issue 2

Full text available:  [pdf\(861.44 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

13 An  $O(n \log n)$  algorithm for the maximum agreement subtree problem for binary trees

Richard Cole, Ramesh Hariharan

January 1996 **Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms**

Full text available:  [pdf\(1.03 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

14 Maximum  $k$ -chains in planar point sets: combinatorial structure and algorithms


Stefan Felsner, Lorenz Wernisch

June 1993 **Proceedings of the twenty-fifth annual ACM symposium on Theory of computing**

Full text available:  [pdf\(760.00 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

15 Good splitters for counting points in triangles

J. Matoušek, E. Welzl

June 1989 **Proceedings of the fifth annual symposium on Computational geometry**Full text available:  [pdf\(653.71 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A set  $A$  of  $n$  points in the plane has to be stored in such a way that for any query triangle  $t$  the number of points of  $A$  inside  $t$  can be computed efficiently. For this problem a solution is presented with  $O(\sqrt{n} \log n)$  query time,  $O(n \log n)$  space and  $O(n)$

# 16 Algorithms on Stings, Trees, and Sequences: Computer Science and Computational Biology

Dan Gusfield

December 1997 **ACM SIGACT News**, Volume 28 Issue 4Full text available:  [pdf\(1.20 MB\)](#) Additional Information: [full citation](#)

# 17 A work-optimal CGM algorithm for the LIS problem

Garcia Thierry, Myoupo Jean-Frédéric, Semé David

July 2001 **Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures**Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper presents a work-optimal CGM algorithm that solves the Longest Increasing Subsequence Problem. It can be implemented in the CGM with  $P$  processors in  $O(N^2 \div P)$  time and  $O(P)$  communication steps. It is the first CGM algorithm for this problem and it is work-optimal since the sequential algorithm has a complexity of  $O(N^2)$ .

# 18 An $O(n)$ algorithm for determining a near-optimal computation order of matrix chain products

Francis Y. Chin

July 1978 **Communications of the ACM**, Volume 21 Issue 7Full text available:  [pdf\(575.28 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper discusses the computation of matrix chain products of the form  $M_1 \times M_2 \times \dots \times M_n$  where  $M_i$ 's are matrices. The order in which the matrices are computed affects the number of operations. A sufficient condition about the association of the matrices in the optimal order is presented. An

**Keywords:** approximate algorithm, heuristic algorithm, matrix chain product, matrix multiplication

# 19 Spot-checkers

Funda Ergün, Sampath Kannan, S. Ravi Kumar, Ronitt Rubinfeld, Mahesh Viswanathan

May 1998 **Proceedings of the thirtieth annual ACM symposium on Theory of computing**Full text available:  [pdf\(1.33 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

# 20 On the general graph embedding problem with applications to circuit layout

Jonathan S. Turner

January 1984 **ACM SIGACT News**, Volume 16 Issue 1Full text available:  [pdf\(63.39 KB\)](#) Additional Information: [full citation](#), [abstract](#)

This paper addresses a class of graph embedding problems in which the object is to map the vertices of one graph to the vertices of another, so that the images of vertices that are adjacent in the  $\langle u \rangle$  source graph are close together in the  $\langle u \rangle$  target graph. An

important special case is the bandwidth minimization problem in which the target graph is a simple line graph. In a previous paper, this author showed that for random graphs having bandwidth at most  $k$  ...

Results 1 - 20 of 20

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used [code blocks](#) [movement](#) [new](#) [original](#) [index](#) [value](#)

Found 23 of 498 searched out of 498.

 Sort results  
by

 Display  
results


[Save results to a Binder](#)

[Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

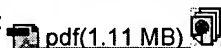
Results 1 - 20 of 23

 Result page: [1](#) [2](#) [next](#)

 Relevance scale ☐ ☐ ☐ ☐ ☐

- 1 [Compilation: Mapping reference code to irregular DSPs within the retargetable, optimizing compiler COGEN\(T\)](#)
- Gary William Gréwal, Charles Thomas Wilson  
December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:


[Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Generating high quality code for embedded processors is made difficult by irregular architectures and highly encoded parallel instructions. Rather than deal with the target machine at every stage of the compilation, a promising new methodology employs generic algorithms to optimize code for an idealized abstraction of the true target machine. This code, called reference code, is then mapped to the real instruction set by enhanced genetic algorithms. One perturbs the original schedule to find a n ...

- 2 [Mimic: a fast system/370 simulator](#)
- C. May  
July 1987 **ACM SIGPLAN Notices , Papers of the Symposium on Interpreters and interpretive techniques**, Volume 22 Issue 7

Full text available: pdf(1.16 MB)

 Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Software simulation of one computer on another tends to be slow. Traditional simulators typically execute about 100 instructions on the host machine per instruction simulated. Newer simulators reduce the expansion factor to about 10, by saving and reusing translations of individual instructions. This paper describes an experimental simulator which takes the progression one step further, translating groups of instructions as a unit. This approach, combined with flow analysis, reduces the expansio ...

- 3 [MPEG-4: an object-based multimedia coding standard supporting mobile applications](#)
- Atul Puri, Alexandros Eleftheriadis  
June 1998 **Mobile Networks and Applications**, Volume 3 Issue 1

Full text available: pdf(747.80 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The ISO MPEG committee, after successful completion of the MPEG-1 and the MPEG-2 standards is currently working on MPEG-4, the third MPEG standard. Originally, MPEG-4 was conceived to be a standard for coding of limited complexity audio-visual scenes at very low bit-rates; however, in July 1994, its scope was expanded to include coding of scenes as a collection of individual audio-visual objects and enabling a range of advanced functionalities not supported by other standards. One of the ke ...

### A history of the SNOBOL programming languages

Ralph E. Griswold

January 1978 **ACM SIGPLAN Notices , The first ACM SIGPLAN conference on History of programming languages**, Volume 13 Issue 8

Full text available:  [pdf\(3.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Development of the SNOBOL language began in 1962. It was followed by SNOBOL2, SNOBOL3, and SNOBOL4. Except for SNOBOL2 and SNOBOL3 (which were closely related), the others differ substantially and hence are more properly considered separate languages than versions of one language. In this paper historical emphasis is placed on the original language, SNOBOL, although important aspects of the subsequent languages are covered.

### 5 A multiple processor data flow machine that supports generalized procedures

Arvind, Vinod Kathail

May 1981 **Proceedings of the 8th annual symposium on Computer Architecture**

Full text available:  [pdf\(725.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Programs for data flow machines are written in functional languages, some of which require efficient support for dynamic procedure invocation to achieve high performance and programming flexibility. Among the proposed data flow machines, few support procedures in any generality. Our machine, which is a hardware realization of the U-interpretter for data flow languages, provides support for a variety of procedure calling conventions. Because the U-interpretter assigns a unique activity name to ...

### 6 Mobile objects in distributed Oz

Peter Van Roy, Seif Haridi, Per Brand, Gert Smolka, Michael Mehl, Ralf Scheidhauer

September 1997 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 19 Issue 5

Full text available:  [pdf\(484.83 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Some of the most difficult questions to answer when designing a distributed application are related to mobility: what information to transfer between sites and when and how to transfer it. Network-transparent distribution, the property that a program's behavior is independent of how it is partitioned among sites, does not directly address these questions. Therefore we propose to extend all language entities with a network behavior that enables efficient distributed programm ...

**Keywords:** latency tolerance, mobile objects, network transparency

### 7 Self-regulation of workload in the Manchester Data-Flow computer

John R. Gurd, David F. Snelling

December 1995 **Proceedings of the 28th annual international symposium on Microarchitecture**

Full text available:  [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

### 8 Evaluation of compiler optimizations for Fortran D on MIMD distributed memory machines

Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng

August 1992 **Proceedings of the 6th international conference on Supercomputing**

Full text available:  [pdf\(1.74 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


The Fortran D compiler uses data decomposition specifications to automatically translate Fortran programs for execution on MIMD distributed-memory machines. This paper introduces and classifies a number of advanced optimizations needed to achieve acceptable performance; they are analyzed and empirically evaluated for stencil computations.

Profitability formulas are derived for each optimization. Results show that exploiting parallelism for pipelined computations, reductions, and scans is vi ...

### 9 Prototyping Fortran-90 compilers for massively parallel machines

Marina Chen, James Cowie

July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation**, Volume 27 Issue 7

Full text available:  pdf(1.12 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Massively parallel architectures, and the languages used to program them, are among both the most difficult and the most rapidly-changing subjects for compilation. This has created a demand for new compiler prototyping technologies that allow novel style of compilation and optimization to be tested in a reasonable amount of time. Using formal specification techniques, we have produced a data-parallel Fortran-90 subset compiler for Thinking Machines' Connection Machine/2 and Conne ...

### 10 Locality phase prediction

Xipeng Shen, Yutao Zhong, Chen Ding

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 38 , 39 , 32 Issue 5 , 11 , 5

Full text available:  pdf(739.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As computer memory hierarchy becomes adaptive, its performance increasingly depends on forecasting the dynamic program locality. This paper presents a method that predicts the locality phases of a program by a combination of locality profiling and run-time prediction. By profiling a training input, it identifies locality phases by sifting through all accesses to all data elements using variable-distance sampling, wavelet filtering, and optimal phase partitioning. It then constructs a phase hiera ...

**Keywords:** dynamic optimization, locality analysis and optimization, phase hierarchy, program phase analysis and prediction, reconfigurable architecture

### 11 Automatic array alignment in data-parallel programs

Siddhartha Chatterjee, John R. Gilbert, Robert Schreiber, Shang-Hua Teng

March 1993 **Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available:  pdf(1.34 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Data-parallel languages like Fortran 90 express parallelism in the form of operations on data aggregates such as arrays. Misalignment of the operands of an array operation can reduce program performance on a distributed-memory parallel machine by requiring nonlocal data accesses. Determining array alignments that reduce communication is therefore a key issue in compiling such languages. We present a framework for the automatic determination of array alignments in data-parallel la ...

### 12 Novel ideas: A design space evaluation of grid processor architectures

Ramadass Nagarajan, Karthikeyan Sankaralingam, Doug Burger, Stephen W. Keckler

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  pdf(1.29 MB)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)  
[Publisher Site](#)

In this paper, we survey the design space of a new class of architectures called Grid Processor Architectures (GPAs). These architectures are designed to scale with technology, allowing faster clock rates than conventional architectures while providing superior instruction-level parallelism on traditional workloads and high performance across a range of application classes. A GPA consists of an array of ALUs, each with limited control,

connected by a thin operand network. Programs are executed b ...

**13 MEDEA workshop: An EGA approach to the compile-time assignment of data to multiple memories in digital-signal processors**

Gary Gréwal, Tom Wilson, Andrew Morton

March 2003 **ACM SIGARCH Computer Architecture News**, Volume 31 Issue 1

Full text available:  [pdf\(1.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we present a methodology, based on an Enhanced Genetic Algorithm (EGA), for assigning data objects to dual-bank memories. Our approach is global, and special effort is made to identify those objects that could potentially benefit from an assignment to a specific memory, or perhaps duplication in both memories. The enhancements to the genetic algorithm include a directed mutation operator and a new type of elitism. Together, these enhancements improve the performance of the genetic ...

**14 Extend: a library-based, hierarchical, multi-domain modeling system**

Bob Diamond

December 1993 **Proceedings of the 25th conference on Winter simulation**

Full text available:  [pdf\(811.39 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)

**15 The Dynamic Incremental Compiler of APL\3000**

Ronald L. Johnston

May 1979 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL: part 1**, Volume 9 Issue 4

Full text available:  [pdf\(462.40 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Most APL implementations to date have been interpretive because of the dynamic nature of the language. APL\3000 employs a Dynamic Incremental Compiler to allow all the flexibility of change afforded by interpretation, but giving the added bonus of faster execution for programs run more than once. APL\3000 compiles code on a statement-by-statement basis as needed, saving the code and reusing it where possible. A statement is recompiled only when made necessary by changes in syntax or changes ...

**16 Columns: Risks to the public in computers and related systems**

Peter G. Neumann


March 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 2

Full text available:  [pdf\(1.54 MB\)](#) Additional Information: [full citation](#)

**17 Dynamic data distribution with control flow analysis**

Jordi Garcia, Eduard Ayguade, Jesus Labarta

November 1996 **Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM) - Volume 00**

Full text available:  [html\(44.21 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the design of a data distribution tool which automatically derives the data mapping for the arrays and the parallelization strategy for the loops in a Fortran 77 program. The layout generated can be static or dynamic, and the distribution is one-dimensional BLOCK or CYCLIC. The tool takes into account the control flow statements in the code in order to better estimate the behavior of the program. All the information regarding data movement and parallelism is contained i ...

**18 A parallel pipelined processor with conditional instruction execution**

Rod Adams, Gordon Steven

March 1991 **ACM SIGARCH Computer Architecture News**, Volume 19 Issue 1

Full text available:  [pdf\(602.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

In a recent paper by Smith, Lam and Horowitz [1] the concept of 'boosting' was introduced, where instructions from one of the possible instruction streams following a conditional branch were scheduled by the compiler for execution in the basic block containing the branch itself. This paper describes how code from both instruction streams following a conditional branch can be considered for execution in the basic block containing the branch. Branch conditions are stored in Boolean registers and a ...

#### 19 Recency-based TLB preloading

Ashley Saulsbury, Fredrik Dahlgren, Per Stenström

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available:  [pdf\(651.05 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Caching and other latency tolerating techniques have been quite successful in maintaining high memory system performance for general purpose processors. However, TLB misses have become a serious bottleneck as working sets are growing beyond the capacity of TLBs. This work presents one of the first attempts to hide TLB miss latency by using preloading techniques. We present results for traditional next-page TLB miss preloading - an approach shown to cut so ...

#### 20 Technical contributions: "Structured programming" considered harmful

Paul Abrahams

April 1975 **ACM SIGPLAN Notices**, Volume 10 Issue 4

Full text available:  [pdf\(948.53 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)

Results 1 - 20 of 23

Result page: [1](#) [2](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)





USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used [code blocks](#) [movement](#) [new](#) [original](#) [index](#) [value](#)

Found 23 of 498

Sort results by

[Save results to a Binder](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)

Display results

[Search Tips](#)
☐ Open results in a new window

Results 21 - 23 of 23

Result page: [previous](#) [1](#) [2](#)Relevance scale ☐ ☐ ☐ ☐ ☐**21 [A high-speed message-driven communication architecture](#)**

J. Peterson, E. Chow, H. Madan

June 1988 **Proceedings of the 2nd international conference on Supercomputing**Full text available: [pdf\(1.27 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The performance of a message-passing multiple instruction multiple data (MIMD) concurrent computer depends in large part on the communication processing overhead. A high-speed communication architecture is proposed for a hypercube-type supercomputer to attain the specific goals of message-driven processing. These goals include: direct hardware execution of messages, queueing of messages (using various paradigms), adaptive message routing, and special local registers for fast context ...

**22 [Exploiting ILP, TLP, and DLP with the polymorphous TRIPS architecture](#)**

Karthikeyan Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh, Doug Burger, Stephen W. Keckler, Charles R. Moore

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2Full text available: [pdf\(219.05 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper describes the **polymorphous** TRIPS architecture which can be configured for different granularities and types of parallelism. TRIPS contains mechanisms that enable the processing cores and the on-chip memory system to be configured and combined in different modes for instruction, data, or thread-level parallelism. To adapt to small and large-grain concurrency, the TRIPS architecture contains four out-of-order, 16-wide-issue Grid Processor cores, which can be partitioned when easi ...

**23 [Performance of the CRAY T3E multiprocessor](#)**

Ed Anderson, Jeff Brooks, Charles Grassl, Steve Scott

November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**Full text available: [pdf\(89.00 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The CRAY T3E is a scalable shared-memory multiprocessor based on the DEC Alpha 21164 microprocessor. The system includes a number of novel architectural features designed to tolerate latency, enhance scalability, and deliver high performance on scientific and engineering codes. Included among these are **stream buffers**, which detect and prefetch down small-stride reference streams, **E-registers**, which provide latency hiding and non-unit-stride access capabilities, barrier and fetch\_an ...

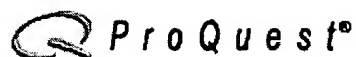
Results 21 - 23 of 23

Result page: [previous](#) [1](#) [2](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Interface language:

English

Databases selected: Multiple databases...

[What's new](#)

## Results

6 documents found for: "longest increasing subsequence" [Set up Alert](#) [About](#)☐ Mark all 0 marked items: Email / Cite / Export [Show only full text](#)Sort results by: [Most recent first](#)

- 
- ☐ 1. **Hammersley's process with sources and sinks**  
*Eric Cator, Piet Groeneboom. Annals of Probability.* Hayward: May 2005. Vol. 33, Iss. 3; p. 879  
[Abstract](#)
- 
- ☐ 2. **Alignment of BLAST high-scoring segment pairs based on the longest increasing subsequence algorithm**  
*Hongyu Zhang. Bioinformatics.* Oxford: Jul 22, 2003. Vol. 19, Iss. 11; p. 1391  
[Article image - PDF](#) [Abstract](#)
- 
- ☐ 3. **Alignment of BLAST high-scoring segment pairs based on the longest increasing subsequence algorithm.**  
*Zhang H. Bioinformatics [NLM - MEDLINE].* Jul 22 2003. Vol. 19, Iss. 11; p. 1391  
[Article image - PDF](#) [Abstract](#)
- 
- ☐ 4. **Recent progress in algebraic combinatorics**  
*Richard P Stanley. American Mathematical Society. Bulletin, New Series, of the American Mathematical Society.* Providence: Jan 2003. Vol. 40, Iss. 1; p. 55  
[Abstract](#)
- 
- ☐ 5. **Enumerating longest increasing subsequences and patience sorting**  
*Sergei Bespamyatnikh, Michael Segal. Information Processing Letters.* Amsterdam: Nov 30, 2000. Vol. 76, Iss. 1-2; p. 7  
[Abstract](#)
- 
- ☐ 6. **Longest increasing subsequences: From patience sorting to the Baik-Deift-Johansson theorem**  
*Aldous, David, Diaconis, Persi. American Mathematical Society. Bulletin, New Series, of the American Mathematical Society.* Providence: Oct 1999. Vol. 36, Iss. 4; p. 413 (20 pages)  
[Abstract](#)
- 

1-6 of 6

Want an alert for new results sent by email? [Set up Alert](#) [About](#)Results per page: [30](#)

## Basic Search

[Tools:](#) [Search Tips](#) [Browse Topics](#) [2 Recent Searches](#)[Search](#)[Clear](#)Database: [Multiple databases...](#) [Select multiple databases](#)Date range: [All dates](#)Limit results to: ☐ Full text documents only

☐ Scholarly journals, including peer-reviewed  [About](#)

 [More Search Options](#)

Copyright © 2005 ProQuest Information and Learning Company. All rights reserved. [Terms and Conditions](#)

[Text-only interface](#)





Welcome United States Patent and Trademark Office

Search Results

[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)[SUPPORT](#)

Results for "'('sequential data compression')&lt;in&gt;metadata)'"

Your search matched 6 of 1235066 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

## » Search Options

[View Session History](#)[New Search](#)

## Modify Search


☐ Check to search only within this results set

 Display Format: ☒ Citation ☐ Citation & Abstract

## » Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

## Select Article Information

- ☒ 1. **A universal algorithm for sequential data compression**  
 Ziv, J.; Lempel, A.;  
 Information Theory, IEEE Transactions on  
 Volume 23, Issue 3, May 1977 Page(s):337 - 343  
[AbstractPlus](#) | Full Text: [PDF](#)(1016 KB) IEEE JNL
- ☐ 2. **Improved variations relating the Ziv-Lempel and Welch-type algorithms for sequential data compression**  
 Yokoo, H.;  
 Information Theory, IEEE Transactions on  
 Volume 38, Issue 1, Jan. 1992 Page(s):73 - 81  
 Digital Object Identifier 10.1109/18.108251  
[AbstractPlus](#) | Full Text: [PDF](#)(588 KB) IEEE JNL
- ☐ 3. **A two-level N-ary tree based sequential data compression algorithm**  
 Guobin Shen; Luxi Yang; Yimin Mao;  
 Signal Processing, 1996., 3rd International Conference on  
 Volume 1, 14-18 Oct. 1996 Page(s):702 - 705 vol.1  
 Digital Object Identifier 10.1109/ICSIGP.1996.567360  
[AbstractPlus](#) | Full Text: [PDF](#)(296 KB) IEEE CNF
- ☐ 4. **Context weighting for general finite-context sources**  
 Willems, F.M.J.; Shtarkov, Y.M.; Tjalkens, T.J.;  
 Information Theory, IEEE Transactions on  
 Volume 42, Issue 5, Sept. 1996 Page(s):1514 - 1520  
 Digital Object Identifier 10.1109/18.532891  
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(588 KB) IEEE JNL
- ☐ 5. **Online suffix trees with counts**  
 Nuallain, B.O.; de Rooij, S.;  
 Data Compression Conference, 2004. Proceedings. DCC 2004  
 23-25 March 2004 Page(s):555  
 Digital Object Identifier 10.1109/DCC.2004.1281531  
[AbstractPlus](#) | Full Text: [PDF](#)(220 KB) IEEE CNF
- ☐ 6. **Data compression for CAM with weighted double tracing method**  
 Fujimoto, M.; Kariya, K.;  
 Industrial Electronics, Control and Instrumentation, 1994. IECON '94., 20th International Conference on  
 Volume 2, 5-9 Sept. 1994 Page(s):1171 - 1175 vol.2  
 Digital Object Identifier 10.1109/IECON.1994.397957

*cited in IDS*

[AbstractPlus](#) | Full Text: [PDF\(284 KB\)](#) [IEEE CNF](#)



Indexed by  
 Inspec

[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)

© Copyright 2005 IEEE – All Rights Reserved